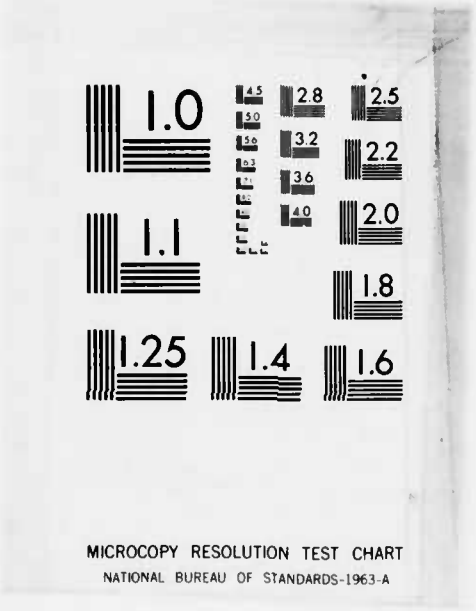
AD-A132 164 ARTIFICIAL INTELLIGENCE TECHNIQUES FOR INDUSTRIAL APPLICATIONS IN JOB SHOP SCHEDULING(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA W B TOWNSEND JUN 83 1/1

UNCLASSIFIED F/G 9/2 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(2)

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

ARTIFICIAL INTELLIGENCE TECHNIQUES
FOR INDUSTRIAL APPLICATIONS IN JOB SHOP SCHEDULING

by

Wade Benton Townsend

June 1983

Thesis Advisor: Philip Bromiley

Approved for public release; distribution unlimited

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A132 164	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Artificial Intelligence Techniques for Industrial Applications in Job Shop Scheduling		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis June 1983
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Wade Benton Townsend		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE June 1983
		13. NUMBER OF PAGES 47
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Artificial Intelligence, Job Shop Scheduling, Frame Representation, Heuristic Search, Expert System, Knowledge Base, Rule Base.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The application of AI (artificial intelligence) techniques to the scheduling of industrial production operations offers a promising new approach to a scheduling problem of great magnitude and complexity. Foremost among these techniques is a powerful knowledge representation language that is capable of modeling the production environment at all levels of detail. The capturing of such complexity in the data base enables the computer to generate feasible schedules from a very large solution space which are highly rated

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

BLOCK 20: ABSTRACT (Continued)

by human experts. An introduction to artificial intelligence is presented that discusses knowledge representation techniques and describes an intelligent scheduling system. The relevance of AI techniques to military industrial production operations is explored by examining the closed job shop in the context of jet engine repair.

Approved for public release; distribution unlimited

Artificial Intelligence Techniques for Industrial Applications in Job Shop Scheduling

by

Wade Benton Townsend
Lieutenant, United States Navy
B.S., Florida State University, 1976

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
June 1983

Author: Wade B. Townsend

Approved by: _____ Thesis Advisor

_____ Second Reader

_____ Chairman, Department of Administrative Sciences

_____ Dean of Information and Policy Sciences

ABSTRACT

The application of AI (artificial intelligence) techniques to the scheduling of industrial production operations offers a promising new approach to a scheduling problem of great magnitude and complexity. Foremost among these techniques is a powerful knowledge representation language that is capable of modeling the production environment at all levels of detail. The capturing of such complexity in the data base enables the computer to generate feasible schedules from a very large solution space which are highly rated by human experts.

An introduction to artifical intelligence is presented that discusses knowledge representation techniques and describes an intelligent scheduling system. The relevance of AI techniques to military industrial production operations is explored by examining the closed job shop in the context of jet engine repair.

TABLE OF CONTENTS

I. INTRODUCTION

The job shop scheduling problem has been rather extensively treated in the literature and remains a difficult problem with which to cope on both a theoretical and practical level. With the advent of highly reliable integrated circuits and the rapid decline in the cost of computing power, research in artificial intelligence is beginning to produce commercially useful scheduling systems that promise to deal more effectively with the combinatorial aspect of the scheduling problem. One such commercial application was developed at Carnegie-Mellon University by Dr. Mark Fox and his colleagues. The Intelligent Scheduling and Information System (ISIS) is a knowledge based job shop scheduling system that uses domain specific information to guide schedule construction.

Most job shop scheduling applications, including ISIS, have been manufacturing applications. Repair operations add a new wrinkle to an already complicated scheduling problem in that the nature and extent of damages frequently cannot be determined until after the repair process is initiated. In this respect, repair operations stand to gain more from improved scheduling efficiency (e.g., reduced work-in-process inventory levels) than do manufacturing operations. The

prospect of artificial intelligence applications for repair operations seems to be the most promising for closed shops due to a relatively stable and predictable product line.

Chapter II describes the job shop scheduling problem in relation to the scheduling process as a whole and in the context of different production processes (i.e., manufacturing versus repair) and shop types (i.e., open versus closed). Solution techniques are reviewed briefly, focusing on queueing theory and computer simulation.

Chapter III provides an overview of artificial intelligence of an introductory level and describes an intelligent scheduling system. A discussion of production systems, search methods and frame representations supports a general description of CMU's ISIS.

Chapter IV offers conclusions and recommendations concerning the relevancy of intelligent scheduling systems to repair operations.

II. THE JOB SHOP SCHEDULING PROBLEM

There are two general types of scheduling--aggregate and detailed. The former plans for the overall level of output for a production system and procures the required inputs, while the latter allocates those inputs. Aggregate planning (master scheduling) usually involves time periods of three months to a year, whereas detailed scheduling deals with day-to-day operations.

Job shop scheduling is normally associated with detailed scheduling in that it is primarily concerned with the loading and sequencing of orders. Loading refers to the allocation of jobs to processing or work centers. Sequencing establishes the order (priority) in which jobs will be processed by a work center. The end product of the scheduling process is the final detailed schedule which specifies starting or completion times for jobs at each work center. Figure 1 illustrates the relationship of these activities to the scheduling process as a whole.

Job shops are intermittent production systems. The essential distinguishing characteristic of an intermittent system is that it produces a variety of products to customer order. Because each order usually requires distinctive processing, production control is tantamount to order control, and is so called. The objective of order control is to

AGGREGATE SCHEDULING	DETAILED SCHEDULING		
DEVELOPING MASTER SCHEDULE	—> LOADING	—> SEQUENCING	—> DEVELOPING FINAL SCHEDULE
Specify overall level of output for planning period	Assign jobs to work centers	Specify processing sequence of all jobs at each work center	Specify start/finish, dates/times of all jobs at each work center

Figure 1. The Scheduling Process

process each order efficiently as it moves through the facility [Ref. 1: p.493]. That objective is accomplished by allocating resources (i.e., machines, materials and personnel-- the factors of production) according to time (due date) and place (routing) constraints. The allocation of resources in this manner is the purpose of scheduling [Ref. 2: p. 375].

Schedule construction requires detailed information about the job itself, as well as information on the availability of the factors of production. Each job requires one or more operations which have specified precedence relations that determine the technological ordering or routing of a job to various processing centers. This information is available from route sheets and is the basis for shop loading (i.e., jobs are allocated to work centers based on the processing required by each operation and the ordering of operations). Each operation is described on the operation sheet, from

which an estimate of the job processing time can be made based on the indicated scope of the work to be done. Various methods may then be used for shop loading. Shop loading is usually a problem when more than one work center has the capability to process a job and a preference must be expressed as to which work center will process it. Performance criterion such as shortest processing time (SPT) or lowest cost are used to evaluate each work center's performance. Job assignments are ultimately made on the basis of an individual work center's performance, subject to capacity limitations.

Once the jobs are loaded, they must be sequenced. Sequencing is accomplished by applying priority rules to those jobs waiting to be processed by the work center. The rules determine which job will be processed next on an idle machine. Given that the composition of waiting jobs changes continuously, priority rules must be designed to favor the movement of orders according to appropriate performance criteria, subject to loading constraints (e.g., minimize the average order flow time). Loading influences the overall performance of priority rules to the extent that an overload or underload exists [Ref. 1: p. 507].

Priority rules are heuristic in the sense that they are practical guidelines that suggest a satisfactory, but sub-optimal sequence. Unfortunately, as the number of jobs and

work centers increase, sequencing rapidly becomes too complex for optimal analytical techniques. The simple sequencing of five jobs on three machines results in $(5!)^3$ or 1,728,000 alternatives. The essence of the job shop scheduling problem is the computational effort associated with problems of sufficient size to be useful in the real world. In such cases, the complete enumeration and evaluation of possible sequences is impractical, even with the aid of a computer. For instance, in the case of five jobs and seven machines there are $(5!)^7$ or approximately 3.58×10^{14} sequences which would require more than 11 years of computer time at the processing rate of a microsecond each. This compares to about 2 seconds of computer time for the first case and illustrates the magnitude and complexity of the sequencing problem. It is desirable, therefore, to reduce the number of paths along which the computer must search for a feasible sequence by some means other than analytical, due to the mathematical complexity or computational effort involved. Heuristic rules selectively prune those paths that appear to lead to inefficient or non-feasible sequences at the risk of eliminating an optimal one, but provide a feasible sequence with reasonable effort.

Scheduling is not the same as sequencing in the sense that scheduling's focus is on the time of events as opposed to the arrangement of events. A schedule is a feasible

sequence of operations that have been assigned starting and completion times, to include idle time between operations, if any [Ref. 3: p. 8]. Developing the final detailed schedule, then, is a matter of establishing starting and completion dates for all jobs based on estimates of processing times and due dates. Schedules may be constructed backwards to meet a required delivery date or forward to produce as soon as possible.

A. THE JOB SHOP

Typically, the term job shop is used in a manufacturing context to describe a shop production system (e.g., the jobbing machine shop whence came the term) in which a variety of jobs are manufactured according to customer specifications. This is the general case in that the shop is open to job orders from anyone, and is distinguished from a closed shop which is not open to outside orders.

The primary attribute of both the open and the closed shop is the functional layout of its equipment. That is, machines are grouped by function on the plant floor. This derives from the nature of the jobs that are being processed. For example, the custom nature of the manufacturing process in the open shop generates different technological, design and material requirements for each order. Under such circumstances, a functional layout promotes the economical utilization of equipment by collecting the fractional usage

demands (i.e., time-sharing) for each job. Also, the dissimilar characteristics of the jobs don't justify a production line layout [Ref. 4: p. 374].

The closed shop is the captive shop of an enterprise and produces to inventory for the internal use of that enterprise in its own product line. The closed shop's product line is, therefore, more predictable in nature than that of an open shop, although captive shops may receive internal one-time orders [Ref. 4: p.373].

Buffa and Taubert [Ref. 4: p. 373] note that much of the available literature assumes that the job shop is typified by the open shop when in fact the closed shop may be more common and easier to deal with.

B. MANUFACTURING VERSUS REPAIR

The job shop is less frequently associated with the repair process than the manufacturing process. Manufacturing and repair both represent a transformation process whereby goods or services are created. That is, <u>they are both production activities</u>. However, manufacturing is inherently more predictable than repair in that the item being produced can be <u>precisely</u> defined in terms of the specific type of manufacturing process, processing times and material requirements. The repair process, on the other hand, varies with the material condition of the item being repaired.

The repair of jet engines for naval aircraft is conducted at Naval Air Rework Facilities (NARFs) and is an important example of a repair process closed shop production system. NARFs produce to inventory for the Navy Supply System.

The rework process is initiated with the induction of a retrograde engine, which is disassembled, repaired, assembled, and tested prior to certification that the engine is ready for issue (RFI). Engine repair is broadly classified as either major or minor. Slaybaugh [Ref. 5: p 11] notes that the main criteria for engine repair classification is the depth of disassembly required to effect repairs (see Figure 2). The repair process can thus be depicted in terms of the engine's hierarchical structure, the depth of which varies for each engine inducted due to differences in material condition. That is, the process of disassembly and assembly peculiar to each repair can be represented internally to the computer and externally to the user by an exploded, level-by-level tree diagram of the engine's internal structure (see Figure 3). This procedure is basic to Material Requirements Planning (MRP), a technique for determining the quantities of components required to

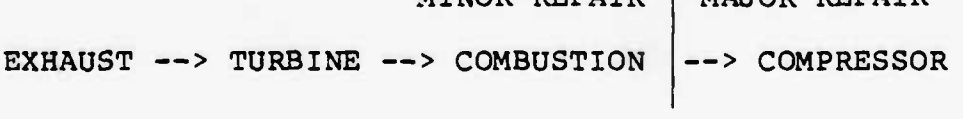


Figure 2. The Repair Process (Sectional Disassembly)

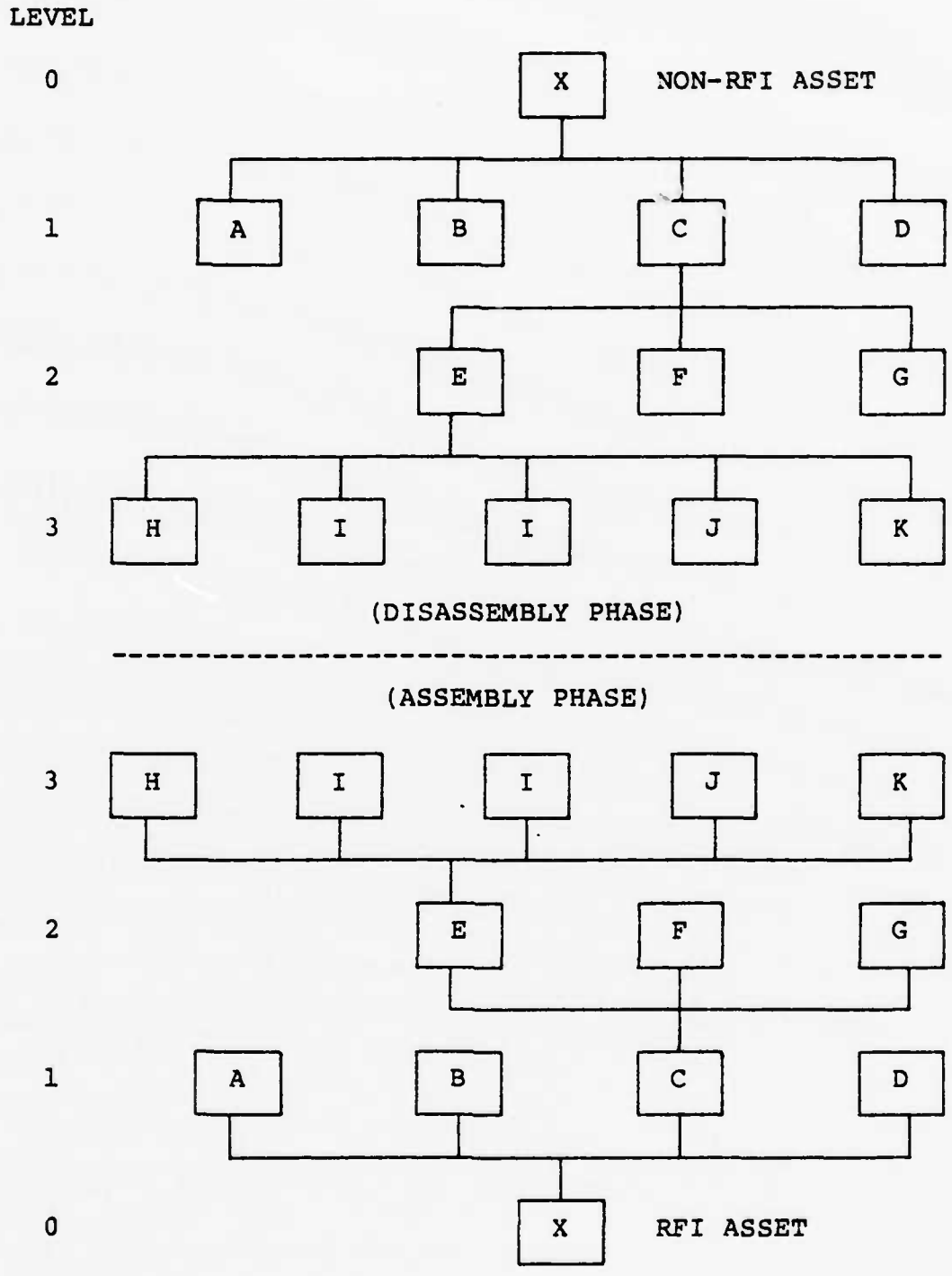


Figure 3. NARF MRP Product Structure [Ref. 5: p. 17]

manufacture a product and may be extended to the repair process in a broad context; namely, production control and scheduling.

C. SOLUTION TECHNIQUES

Optimal analytical methods are of little practical use in solving job shop scheduling problems because they are limited to a relatively small scale. No general solution exists, for example, for sequencing problems with more than two work centers, although there are heuristic approximations of the optimal solution [Ref. 1: p. 511]. The various optimization methods include combinatorial analysis, integer and dynamic programming and network analysis. Although these techniques are important in their own right, the most significant contribution has been made by formulating problems as queueing systems and simulating their solution using various heuristic rules.

The majority of research published in the production literature focuses on the sequencing problem. The impetus for this research was the recognition in the early 1950's that the job shop scheduling problem could be formulated as a system of waiting lines (queues). This realization was significant because it suggested that once the problem was formulated in this manner, the queue discipline (priority or dispatching rules) could be manipulated to provide some insight into the sequencing problem. A queueing system (see

Figure 4) can be described in the context of the job shop scheduling problem by defining the nature of the service process (e.g., repair), the type of service facility (e.g., machine) and the rates at which jobs arrive and are processed. When the arrival rate is larger than the service rate, a waiting line will build up and the queue discipline will determine how jobs are sequenced. The relative effectiveness of various priority rules as measured by the system's performance can be tested by repeatedly solving the problem using different rules.

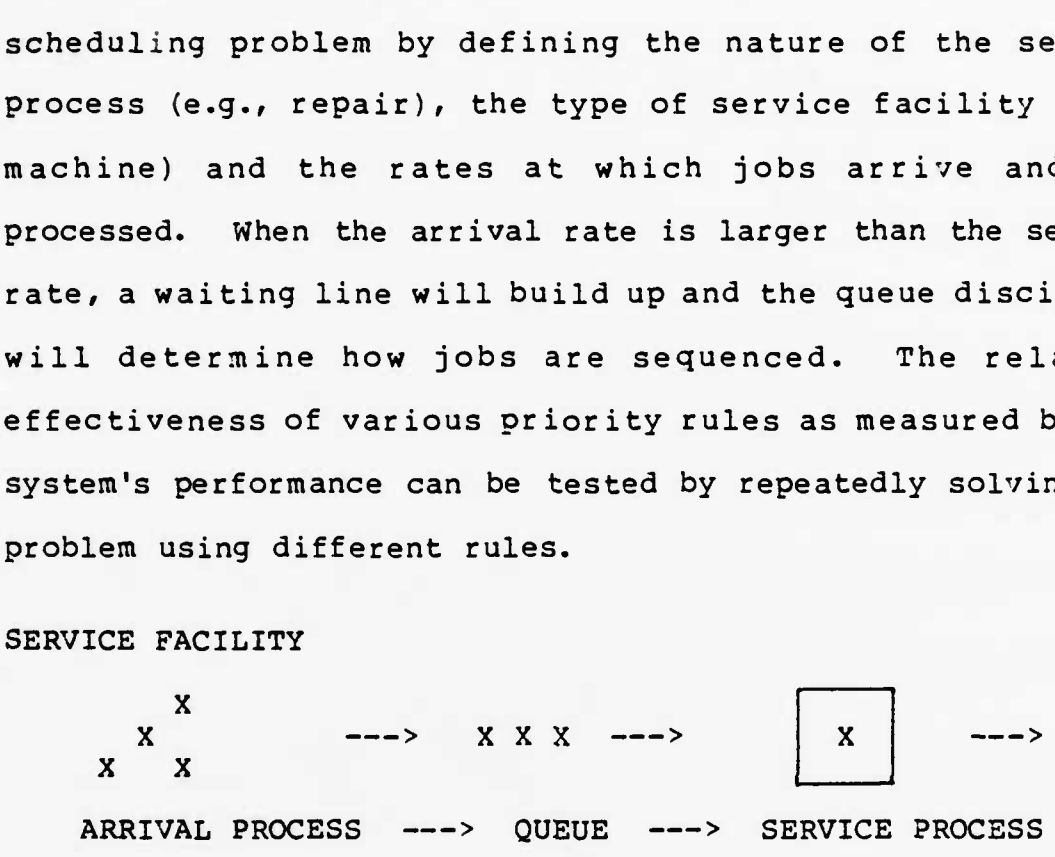


Figure 4. The Elements of a Queueing System

Although this is a viable approach to sequencing problems, queueing theory is severely limited as an analytical solution technique for two reasons. First, the complex job shops encountered in reality rarely, if ever, exhibit the distributions of arrival times and processing times assumed by queueing theory (e.g., exponential and Poisson). Second, the mathematical complexity associated with complex queueing situations found in many job shops becomes very great. Simulation has proven useful in this

regard because it can handle empirical data that does not fit the standard theoretical statistical distributions, and because it can represent complex systems more effectively since its conceptual framework is not limited in a mathematical sense. Thus, for complex systems, computer simulation is an advantageous procedure for evaluating priority rules. The main disadvantage of simulation is the high cost tradeoff associated with modeling such complex systems [Ref. 2: p. 388]. That is, a highly sophisticated model of a system may be very realistic in that it includes many real world variables, but it may be substantially more expensive to build and run than a less sophisticated model.

III. ARTIFICIAL INTELLIGENCE

Programs that endow computers with "artificial intelligence" hold great promise for applications in many relevant areas. The application of AI (artificial intelligence) techniques to the scheduling of industrial production operations offers a promising new approach to a scheduling problem of great magnitude and complexity. Foremost among these techniques is a powerful knowledge representation language that is capable of modeling the production environment at all levels of detail. The capturing of such complexity in the data base enables the computer to generate feasible schedules from a very large solution space which are highly rated by human experts.

This chapter presents an introduction to AI and describes an artificially intelligent scheduling system developed for Westinghouse by the Intelligent Systems Laboratory at Carnegie-Mellon University.

A. BACKGROUND

1. Expert Systems

Difficult decision-making jobs that typically require the judgment and experience of a human expert are being done by AI programs known as knowledge or rule-based expert systems. Until recently, the use of computers was restricted

mainly to high-speed data manipulation done according to rigidly defined algorithmic programs. Expert systems mimic human performance in problem solving by drawing conclusions from task-specific knowledge. This knowledge includes factual or textbook knowledge to be sure, but it also includes the intuition and experiential knowledge of the human expert. All rule-based systems have a collection of rules called the knowledge base, which is a codified representation of this knowledge. In conjunction with a rule interpreter and a global data base of assertions about the case/problem at hand, the knowledge base forms a production system.

a. Production Systems

Production systems solve problems by searching through the space of possible problem states (a.k.a. the state space) that correspond to the condition of the problem at each stage of its solution. In chess, for example, the initial configuration of the playing board represents a problem state from which several alternative states can be generated, depending on the opening move. Each of these new states can be produced by the application of one of the rules for moving chessmen. The object is to modify the problem state progressively, from its initial state to a goal state (i.e., checkmate), by the application of an appropriate

sequence of rules. The term search always refers to the search for a goal achieving sequence [Ref. 6: p. 25].

(1) Production Rules. In a production system, the rules (a.k.a. productions) are inference rules that are based upon what is known about the problem in general. For example, we know that all mammals have hair. If we know that an animal has no hair, then we can infer that it is not a mammal. This simple deduction amounts to a change in the problem state if we are concerned with identifying animal species. That is, what we know about the problem in particular (the data base) now includes <not mammal>, which narrows the focus of the search and advances search progress beyond the initial state. Each of the rules consists of a condition part and an action part, and has the following general form [Ref. 7: p. 246]:

IF: <antecedent$_1$>
.
.
.
<antecedent$_m$>

THEN: <consequent$_1$>
with certainty C_1
.
.
.
<consequent$_n$>
with certainty C_n

If all the antecedents can be matched against assertions in the global data base, then the consequents can be performed. The rule IF: <has no hair> THEN: <is not mammal> specifies a condition (has no hair) that must be present in the global data base before the production can fire (i.e., before the action specified by the consequent can be executed).

(2) The Rule Interpreter and System Operation. The function of the rule interpreter is to examine the production rules to determine which ones are enabled (i.e., capable of being fired) and, after resolving which rule to apply, to fire the selected production. The control strategy of the rule interpreter determines how the enabled rules are found and where to apply them. Forward-chaining is one of the simplest strategies and consists of scanning the rules for matching antecedents, applying the rule found and updating the data base [Ref. 7: 246]. The actions of the fired productions alter the contents of the data base by changing an assertion or by adding a new one (e.g., animal is not mammal) such that other rules may become enabled or disabled. Thus, when a rule is applied, an inference is made and registered in the global data base, thereby generating a new problem state from which the search may be advanced. The process continues until a goal state is reached or no applicable rules are found. This was the strategy followed in the chess example, where the problem state was brought

forward from its initial configuration to a checkmate configuration by the application of an appropriate sequence of rules. Backward-chaining consists of selecting a goal and scanning the rules to find those whose consequent actions can achieve that goal [Ref. 7: p. 246]. Given a goal of integrating $1/(\cos^2 x)dx$, the rule interpreter searches for a rule that will convert the goal expression into one or more simpler expressions that can be integrated. For example, the rule for the identity $1/(\cos x) = \sec x$ converts the goal expression to $\sec^2 x$, for which the immediate solution is $\tan x$ [Ref. 6: p. 23].

b. Knowledge Engineering

In this manner, an expert system draws conclusions through logical or plausible inference, not by calculation. Its effectiveness, therefore, is determined in large measure by the quality of its knowledge base, which is determined by the degree to which domain specific knowledge can be codified into a computer program. The process of interviewing and debriefing the human expert, discovering or elucidating heuristic rules, and codifying both heuristic rules and factual knowledge into a computer program, is called knowledge engineering. Typically, human experts are not very good at remembering special cases until they are confronted with a concrete example, which makes for a rather time-consuming and laborious extraction process.

Fortunately, the knowledge base can be developed incrementally over an extended time by refining old rules and adding new ones [Ref. 7: p. 246].

c. Rule-based Versus Conventional Systems

A major difference between rule-based expert systems and conventional computer programs is the separation of general knowledge about the problem (the rules forming the knowledge base) from the general-reasoning mechanism (the rule interpreter). This partitioning offers several advantages [Ref. 7: pp. 246-248]:

- o The same general system can be used for a variety of applications by replacing the rules for a given domain with rules for another.
- o The same knowledge can be used in different ways (e.g., teaching) by changing the rule interpreter.
- o Separate rules enable the program to explain its reasoning by describing the rules it is applying. This transparent reasoning (so-called in comparison to the opaque or black box quality of an algorithmic computer program), when coupled with a natural language interface, makes expert systems user friendly.
- o The possibility of developing introspective systems that check the consistency of their own rules, and evolutionary ones that can modify existing rules and learn new ones.

2. State-space Search

The new states generated by the sequential application of rules can be represented by a search tree. In this representation, state transformations are depicted as directed arcs between the respective states (see Figure 5).

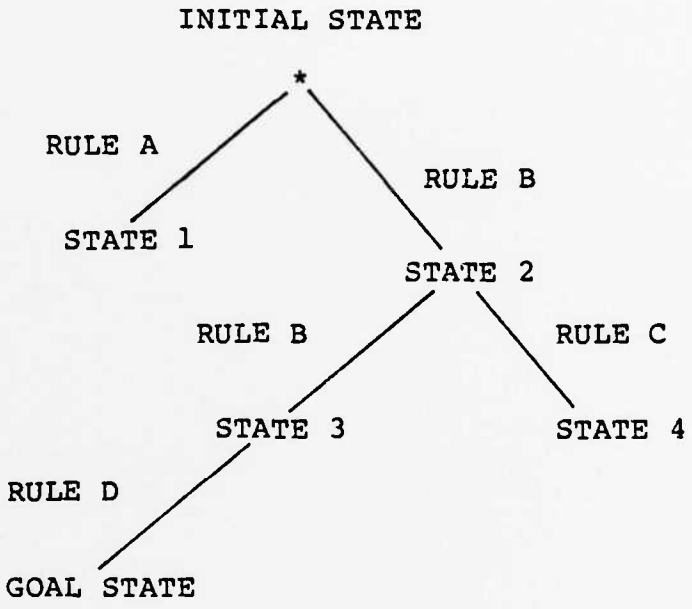


Figure 5. A Search Tree [Ref. 6: p. 34]

The state space being searched (i.e., the search space) may contain one or more paths from the initial state to the goal state. The search objective is to generate a tree just large enough to contain one of these solution paths. A search tree is constructed progressively within the search space and includes only those states for which paths from the initial state have been discovered [Ref. 6: p. 26]. That is, it

includes only that part of the search space that has been made explicit by the sequential application of rules.

Various methods are employed to build search trees that are more or less appropriate to the problem at hand. The kind of problem being addressed in this thesis, however, is characterized by an unimaginably large search space. In chess, for example, the size of the search space grows exponentially with the number of moves and examining all possible sequences of moves is impractical at best [Ref. 6: p. 27]. This problem is called combinatorial explosion and is an important general problem for both operations research and AI applications [Ref. 6: p. 154]. It is desirable, therefore, to limit the search, rather than blindly searching the entire search space. For this reason, exhaustive enumeration techniques that blindly search enormous search spaces are of little practical value in realistic applications. Inasmuch as blind search methods are impracticable for problems of the kind addressed here, we will not consider them further.

One of the most appropriate and important search methods for complex problems is heuristic search. Heuristic search is AI's major contribution to search efficiency in extremely large solution spaces. Whereas blind search arbitrarily explores potential solution paths by generating and testing new problem states until it finds the goal state,

heuristic search uses information about the specific problem domain to judge whether or not a potential solution path is worth pursuing [Ref. 6: pp. 30 and 46]. The current state is said to be expanded when the set of all possible successor states that could result from the application of all applicable rules is generated. Heuristic information can be employed in a search to decide [Ref. 6: p.59]:

1. which state to expand next;
2. which successor state or states to generate during the expansion of the current state; and
3. which states should not be expanded, and eliminated or pruned from the search tree.

An ordered search is an example of the first approach in that the most promising state is expanded next. The promise of a state is estimated by a function called the evaluation function which contains heuristic information about the problem domain.

3. Frames

AI encompasses several areas of research, of which one of the most active is knowledge representation. A representation is a set of conventions about how to describe things [Ref. 8: p. 15]. In general, the more powerful the representation is, the more complete and useful the knowledge base is, and the more effective the system is. Typically, the knowledge base is represented either as production rules

or as frames (a.k.a. schemata) or as a combination of the two representations.

A frame or schema representation is based on the theory that previous situational experiences create certain expectations about objects and events associated with new situations, and provide a framework within which new information can be interpreted. That is, a frame is a structure within which data or knowledge about a stereotyped situation can be represented [Ref. 9: p. 180]. For example, based on previous experience, a chair is generally expected to be a kind of furniture with arms, legs and a back. These expectations represent things that are always true about chairs and provide the context within which other objects can be interpreted. These expectations are represented as terminals or slots within the framework or context of the situation (see Figure 6). The slot provides a mechanism for a kind of reaasoning called expectation-driven processing [Ref. 6: p. 216]. Empty slots (i.e., unconfirmed expectations) can be filled, subject to certain assignment conditions, with data that confirms the expectations. Thus, frame-based reasoning is based on looking for confirmation of expectations and is often just filling in slot values [Ref. 6: p. 219].

One of the important ways in which slot values are specified is by inheritance. In the chair example, the

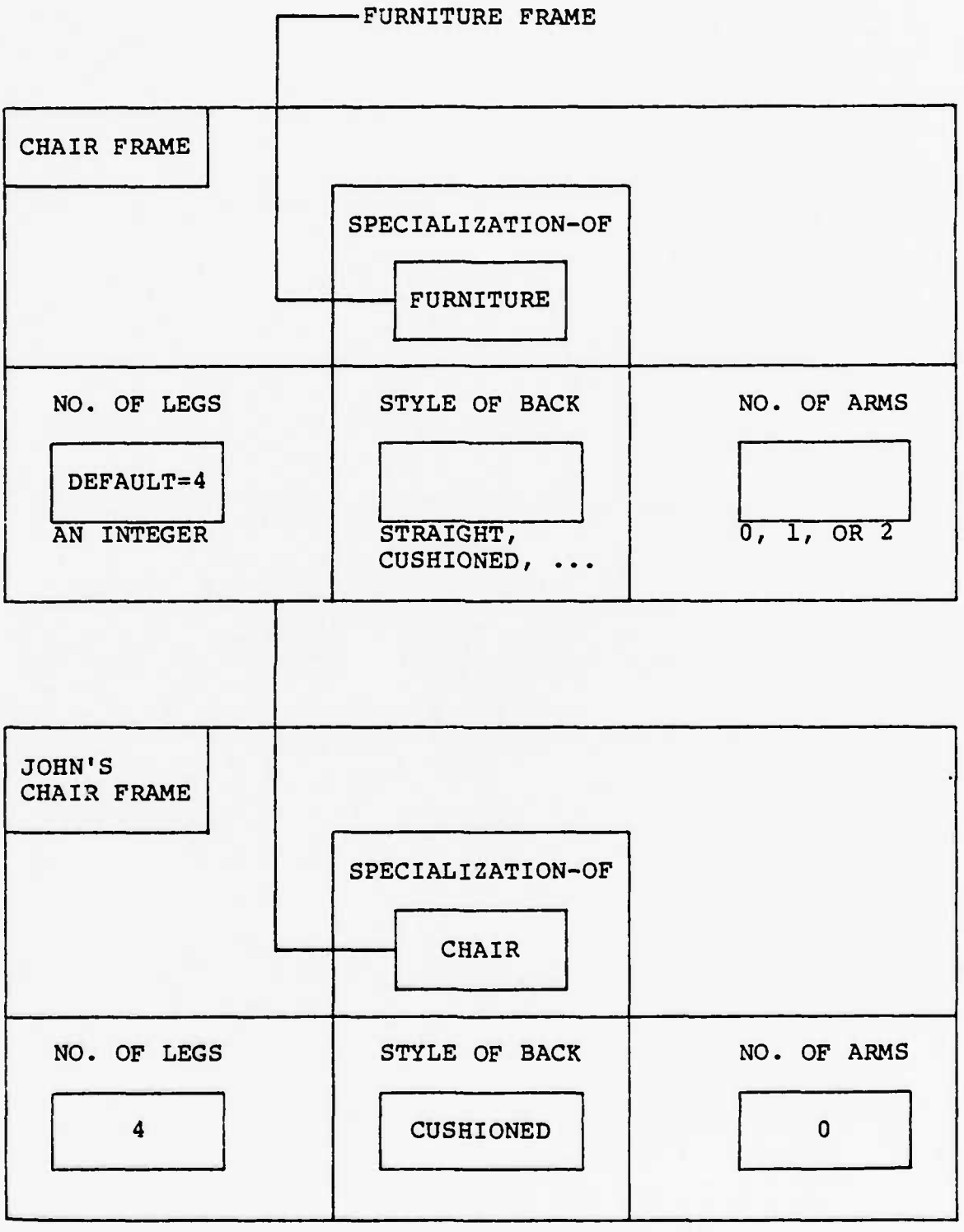


Figure 6. A Chair Frame Representation [Ref. 6: pp. 216-217]

expectation that the chair is a kind of furniture allows information about furniture to be brought to bear on the chair problem. In Figure 6, information in the furniture frame is passed via the specialization-of slot to the chair frame, and the slots from the chair frame (along with their assignment conditions) are inherited by John's chair frame via the specialization-of slot. Information about the parent frame is thus inherited by its children.

Perhaps the simplest way to specify slot values is by default. The default value is attached loosely to the slot so as to be easily displaced by a value that meets the assignment condition. In the absence of information, however, the default value remains attached and expressed.

Finally, procedures can be attached to slots and used to derive slot values. So-called slot-specific heuristics are domain-specific procedures for deriving slot values in a particular context [Ref. 6: p. 220]. An important aspect of attached procedures is that they can be used to direct the reasoning process. In addition to filling in slots, they can be triggered when a slot is filled, possibly altering or terminating the slot filling process [Ref. 6: p. 219].

B. INTELLIGENT SCHEDULING AND INFORMATION SYSTEM (ISIS)

CMU's Intelligent Scheduling and Information System (ISIS) is a knowledge based job shop scheduling system. The prototype of ISIS was developed for a Westinghouse Turbine

Component Plant whose primary product is steam turbine blades. The manufacture of a turbine blade involves a sequence of forging, milling and grinding operations done to close tolerances. The plant produces thousands of different blade styles to order, each of which represents a unique manufacturing process containing at least ten operations. The scheduling problem is to assign starting and completion times to a feasible sequence of operations (i.e., a sequence that does not violate the technological ordering or routing specified by the manufacturing process), and to simultaneously satisfy a large number and variety of constraints.

ISIS represents all relevant scheduling information as constraints and generates feasible schedules within these constraints. In a manner analogous to the construction of a search tree, ISIS generates feasible schedules by articulating several paths (partial schedules) through the search space until a complete path is achieved. After each new state is generated, each partial schedule is rated as to how well it satisfies the constraints determined to be relevant to its path. When conflicting constraints threaten search progress, ISIS relaxes the appropriate constraint according to alternatives specified in the constraints representation. As it works through the search space, ISIS decides which paths to pursue by keeping a set of the best

rated partial schedules. A variety of constraint information is thus brought to bear on the scheduling problem.

1. Constraint Knowledge

In actual practice, schedulers were observed to spend most of their time (80%) communicating with other employees to determine the constraints that were relevant to a particular order's schedule. To the extent that they were unable to determine these constraints, they produced inefficient schedules, which was reflected by the accumulation of work-in-process inventories, low machine utilization and overdue order completions [Ref. 10: pp. 3-4]. Hence, ISIS was designed to perform a heuristic search using information about the job shop scheduling domain expressed as constraints. That is, heuristic information is used to guide or direct the search in an efficient manner.

a. Classification

Four categories of constraints that a scheduler considers were identified [Ref. 11: p. 155]. The first type of constraint is an organizational goal. It rates a scheduling decision according to approximately how well it satisfies the organization's goal of maximizing profits. These constraints include job tardiness, work in process, resource levels, cost, production levels, and shop stability. Physical constraints such as machine limitations are a second category. Gating constraints such as operation precedence

and resource requirements specify conditions to be met before beginning a process or using a resource. Preference constraints, the fourth category, provide the means by which preferences for such things as machines, operations and queue position can be expressed.

b. Frame Representation

In order to be effectively utilized, constraint knowledge must be represented in sufficient detail and in a manner that facilitates the search process. SRL (Schema Representation Language), a frame-based language, is used to model the production environment and to represent the scheduling constraints. Some of SRL's uses follow [Ref. 12: pp. 7-8]:

- o Order definition--Various types of orders.
- o Lot definition--Orders may be grouped into lots and run through the plant as a unit.
- o Resource definition--Resources such as machines, tools, fixtures, materials, and personnel can be defined to include substitutability in operations and current operation assignments.
- o Operation definition--How a product may be produced may be described as an operations graph. The operations graph describes all alternative operations, processing information, and resource requirements. Operations can

be described at varying levels of detail (i.e., hierarchically).

- o Work area definition--Cost centers, work areas, and any other plant floor organizations may be defined.
- o Department definition--Departments, personnel, and any other organization structures may be defined, and linked with other parts of the model.
- o Reservation definition--Any resource may be reserved for an operation.
- o Plant organization--The plant may be described hierarchically, both from an organization structure perspective, and a physical layout perspective.

The general constraint frame contains three slots. The PRECONDITION specifies the applicability of the constraint. The EVALUATION-FUNCTION, when activated, returns a rating of the scheduling decision. The WEIGHT defines the relative importance of the constraint. The representation of constraints also provides information on how to relax a constraint, and the utility of the relaxation [Ref. 11: p. 156].

The decision as to when to relax a constraint is made in two ways [Ref. 11: p. 157]:

- o Generative Relaxation--Constraints are relaxed in generating new alternative states or partial schedules.

- Analytic Relaxation--A rule-based system analyzes each order before it is scheduled to determine the relative importance of constraints and which ones should be relaxed. Another set of rules performs a post-search analysis to determine the reasonableness of the schedule and what other constraints should be relaxed or strengthened, if any.

2. Search

ISIS constructs a schedule by performing a heuristic search. A complete schedule is defined by the path from the initial state to the end state in the search space [Ref. 11: p. 157]. The end state corresponds to the manufactured end item (e.g. a turbine blade). If, for example, the priority-class of the customer order is "forced outage" (i.e., the blade must be shipped by its due date), ISIS bounds the search space by only considering alternative operations, machines and queue positions within the due date constraint. Applying alternative operations, for instance, will generate different intermediate states which correspond to intermediate stages of completion in the manufacturing process. The generated states are rated by applying the constraints. In this manner, ISIS extends the search from initial to intermediate to end states along a path determined by the best rated states.

Post-search analysis is conducted to determine if a satisfactory solution was found. In the turbine blade example, ISIS first attempts to schedule the order backwards from its due date. Failing that, ISIS will schedule forward from the current day. If the resulting schedule is poorly rated, ISIS will redefine the search space by adding the shift dimension for a machine.

3. Source Material Note

The core reference for the preceeding material [Ref. 13] was not available to the author during the research for this thesis. Consequently, the published papers cited and an executive summary of the Fox dissertation provided the materials for the description of ISIS.

IV. CONCLUSIONS AND RECOMMENDATIONS

One of the primary motivations for undertaking this research was to determine what possible relevance ISIS might have for closed repair shops. The NARF Jet Engine Repair Facility, for example, is a closed shop in that the NARF is the designated overhaul point for specific engine models. Since the production environment is relatively well structured and predictable with respect to its product line, the combination of a powerful language capable of representing all possible repair operations, and a heuristic rule base capable of selecting appropriate operations for the repair at hand was particularly interesting. If repair operations could be anticipated with some success, ISIS could prove useful in reducing repair parts inventory.

Another point of interest was the degree to which ISIS might facilitate the integration of production control and inventory control. ISIS can generate trial schedules based upon user specified resource constraints or process specified resource constraints, which provide the production/inventory controller with a "what if?" capability to explore various scheduling alternatives.

A. INVENTORY PLANNING

Scheduling for repair operations in the closed shop is a logical extension of scheduling for manufacturing operations, but one that differs significantly in the degree of certainty to which operations will be known. Much of the uncertainty associated with scheduling repair operations is due to the fact that the nature and extent of damage is frequently not known until after disassembly and inspection. Each repair is unique in that only defective parts are replaced, the quantities and kinds of which vary with the material condition of the item being repaired. Similarly, repair operations vary from job to job. Thus, the selection of operations necessary to effect repair is inherently difficult without information on the nature and extent of damages. However, in most cases, the engine has already been inspected at both the organizational and intermediate level of maintenance before arriving at the depot for rework, in which case pertinent information, however sketchy, is available. In this respect, reliable heuristics could probably be developed from the same diagnostic procedures utilized in troubleshooting engine discrepancies. Of course, the ability of the rule base to anticipate repair operations is related to the quality of information available on the engine, which is probably not too good for the inventory planning period in question (the current or next quarter). The foregoing

suggests that ISIS will probably not be useful for long-range repair shop resource planning. In the short-term, however, ISIS is potentially a very useful repair shop application.

B. MANAGEMENT CONTROL

The ability to exert management control is a function of the usefulness and integrity of the information in the database. The lack of integrity is a serious problem and maintaining it is usually difficult. For example, once an engine is disassembled, its carcass must be strictly controlled and not allowed to float indiscriminantly about the production floor while its components are being reworked. Physical control is necessary to ensure that only changes in engine status authorized by production control take place, and to provide adequate protection for the engine's remaining assets. Repair parts requisitioning must be strictly controlled by engine serial number so that excess or bogus parts are not ordered. Slaybaugh [Ref. 5: p. 29] has documented this practice, which is not a subversive activity of the work force, but rather an expression of the uncertainty inherent in the repair process. These illegitimate stocks buffer workers and work centers from unexpected demand. To a limited extent, ISIS can detect such anomalies when performing constraint-directed scheduling because the information usually contained in the database of database systems is in the constraints. Inconsistencies in

that information will be exposed when ISIS schedules, which provides management with an integrity check.

C. PRODUCTION SCHEDULING

Scheduling engine rework is driven at the component level in that the repair work content above that level is primarily inspection, disassembly and assembly, and testing, with little or no machine processing. Thus, the production/inventory controller must track a large number and variety of components through various work centers in order to maintain accurate status. The engine cannot be reassembled and removed from the work-in-process inventory until all the parts have been replaced or repaired. Improvements in scheduling might be able to reduce the work-in-process inventory (both components and the end item) and improve machine utilization.

Three kinds of in-process improvements are perhaps possible. First, if the repair process on the individual components is not tightly coordinated, one will have large numbers of engines waiting for a final component to be repaired. This is a straight scheduling problem. Second, if different machines can do the same job, improvements in productivity may be feasible by improved schedule construction so as to achieve higher output with the current machines. Third, if there are problems of work shortages for given machines, the scheduling system may be able to identify

which engines should receive top priority so as to raise the output of the entire facility.

In order to achieve these benefits, a day-by-day interactive scheduling system is needed. It is possible that ISIS could be adapted to provide such a support system.

D. INVENTORY OPERATIONS

While the long-term inventory reorder problem is probably best attacked using average production and parts utilization rates, rather than a day-by-day scheduling system, the day-by-day scheduling system may be able to assist inventory management in two very different ways.

First, it would be feasible to build into an ISIS-based scheduling system a preliminary estimate of the parts needed by each work station each day over the planning horizon. This would allow the inventory manager to fine-tune his operation in two ways: (1) by having a warning about short term demand, a look-ahead would be feasible allowing remedial action if shortages were forecast, and (2) inventory workers could begin to work on putting together the next day's orders earlier than they might be able to otherwise. This would allow them to smooth out their workload by filling in slow periods with productive work and would provide quicker response to shop demands.

Second, given a very flexible scheduling system, it might be worth examining the benefits to be derived from

intermittant outage levels. If outages result in extra in-process inventory strewn about the shop floor, and foul-up the schedule and work pattern of the NARF, they are clearly undesirable. On the other hand, if a scheduling system could prospectively adapt to outages by, for instance, not starting repair of engines which would require outaged parts, it might be feasible to lower the parts inventory without damaging production. In short, much of the cost of outages is in fouling up the operations on the floor and if the scheduling system could adapt to avoid such problems, higher outage levels might be feasible.

E. FLEXIBLE SCHEDULING OBJECTIVES

ISIS allows substantial flexibility in the objectives used in making up a schedule. For "forced outages", speed of production and due date deadlines are the most important objectives. Alternatively, manpower/machine utilization, shop stability, average job tardiness, work-in-process inventory, cost, or any of a large spectrum of objectives can be used with whatever relative weights the scheduler wishes. In addition to the desirability of such allowances in making planning analyses, this structure allows for unusual adaptation to circumstances. One obvious example would be in surge capacity. An ISIS scheduling system would be able to accept goals such as "maximum output of engines within the next week" instead of the more common goals. Given the

concern for surge capacity, such scheduling ability might be very desirable.

To summarize, ISIS is an artificially intelligent job shop scheduling system prototype. It has great promise in both manufacturing and repair shop scheduling applications, but has not been tested in day-to-day industrial use. Given its promise, it will be well worth following the system's progress as it moves from research to implementation over the coming years.

LIST OF REFERENCES

1. Tersine, Richard J., Production/Operations Management: Concepts, Structure, and Analysis, Elsevier North Holland, 1980.

2. Adam, Everett E., Jr. and Ebert, Ronald J., Production and Operations Management, 2d ed., Prentice-Hall, 1982.

3. Ashour, S., Sequencing Theory, Lecture Notes in Economics and Mathematical Systems, No. 69, Springer-Verlag, 1972.

4. Buffa, Elwood S. and Taubert, William H., Production-Inventory Systems: Planning and Control, rev. ed., R. D. Irwin, 1972.

5. Slaybaugh, Ernest R., A Preliminary Analysis of TF34-100/400 Jet Engine Rework Data in Support of the MRP System Implementation at NARF Alameda, Master's Thesis, Naval Postgraduate School, Monterey, California, 1981.

6. Barr, Avron and Feigenbaum, Edward A., eds., The Handbook of Artificial Intelligence, Vol. I, W. Kaufman, 1981.

7. Duda, Richard O. and Gaschnig, John G., "Knowledge-Based Expert Systems Come of Age," Byte, September 1981.

8. Winston, Patrick H., Artificial Intelligence, Addison-Wesley, 1977.

9. Marvin Minsky, 1974, as quoted in Patrick Henry Winston, Artificial Intelligence, Addison-Wesley, 1977.

10. Fox, Mark S., et. al., "ISIS: A Constraint-Directed Reasoning Approach to Job Shop Scheduling," Intelligent Systems Laboratory, Robotics Institute, Carnegie-Mellon University, 7 April 1983.

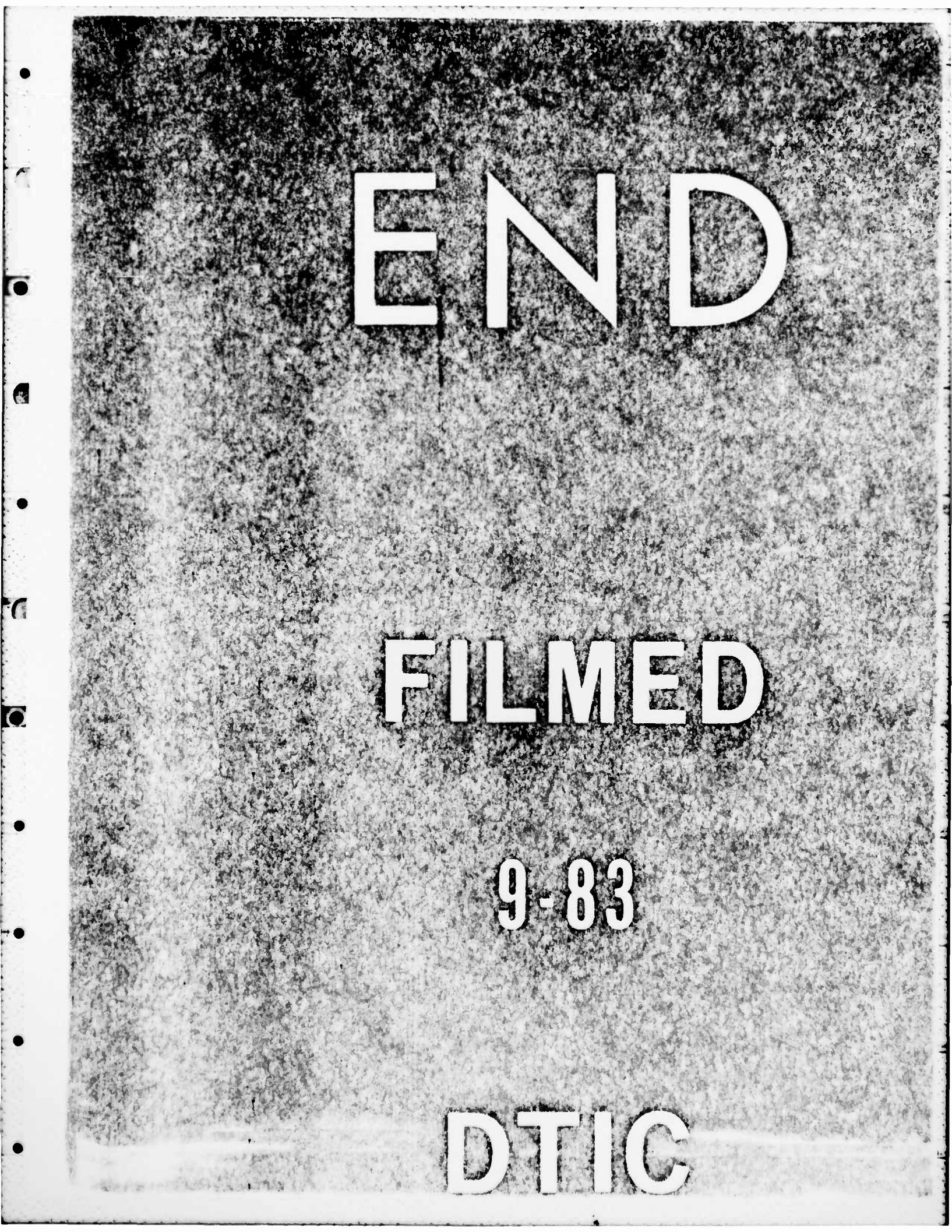
11. Fox, Mark S., Allen, Bradley P. and Strohm, Gary, "Job Shop Scheduling: An Investigation in Constraint-Directed Reasoning," Proceedings of the Second National Conference on Artificial Intelligence, American Association of Artificial Intelligence, August 1982.

12. Fox, Mark S., et. al., "ISIS Production Management and Control System," Project Sampler, Intelligent Systems Laboratory, Robotics Institute, Carnegie-Mellon University, 15 February 1983.

13. Fox, Mark S., Job Shop Scheduling: An Investigation in Constraint-Directed Reasoning, Ph.D. Dissertation, Carnegie-Mellon University, Pittsburgh, Pennsylvania, 1983.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postghraduate School Monterey, California 93940	2
3.	Department Chairman, Code 54Js Department of Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
4.	Mr. David R. Antonelli The Boeing Company P.O. Box 3999 MS 82-91 Seattle, Washington 98124	5
5.	Dr. Philip Bromiley, Code 54Bg Department of Administrative Sciences Naval Postgraduate School Monterey, California 93940	3
6.	Dr. Alan W. McMasters, Code 54Mg Department of Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
7.	Dr. John W. Creighton, Code 54Cf Department of Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
8.	Dr. Dan C. Boger, Code 54Bk Department of Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
9.	LT Wade Benton Townsend, USN ATTN: AIMD Department USS John F. Kennedy (CV-67) FPO New York 09538	5


END
FILMED
9-83
DTIC